REMARKS

On page 2, it is requested that it be clarified how filters are determined simultaneously. It is respectfully submitted that this is fully set forth in the specification of the present application. Initially, it explained that spatial filters may be implemented using a spatial convolution. See lines 1 and 2 of page 2. A spatial convolution calculates what is going on with the pixel brightness around a particular processed pixel. See page 2, lines 2-4.

A spatial convolution may use a mathematical construction of an input pixel and its neighbors to determine an output pixel brightness value. A kernel is a group of pixels used to determine a mathematical calculation of output pixel brightness values. See page 2, lines 15-19. Thus, a convolution may be applied to a kernel to implement a filter. As explained on page 3 of the specification, a kernel convolution may involve a center pixel and, in a 3x3 example, its eight neighbors. The center pixel and its eight neighbors are multiplied by their respective convolution coefficients and the multiplicands may be summed.

As explained at lines 23 and 24 of page 3, conventional convolution masks generally only determine a single kernel size. In some cases, a plurality of different kernels may be needed.

The technique for simultaneously calculating kernels is explained beginning at page 4, line 24. An nxn kernel may be folded along the row and column directions to produce a compacted kernel. This compacted kernel may then be subjected to additions and multiplications in accordance with a spatial filtering algorithm in a way that enables outputs of a variety of different kernel sizes. Thus, a plurality of filters of different sizes may be determined from the same image data.

Referring to Figure 1, the technique for simultaneously calculating the filters is explained at page 7, line 21. Symmetry in the data shown in Figure 2 may be exploited to pre-add pixel data that will be multiplied by the same convolution coefficient. For example, because of symmetry, the diagonals may use the same convolution mask coefficients in one embodiment. Thus, the symmetry may be used to reduce the number of rows of data. In the example shown in the Figure 3, row 11 is added to row 1, row 10 is added to row 2, row 9 is added to row 3, and so on. As a result, the 11x11 matrix of data ends up as a 6x11 kernel.

Referring to Figures 4 and 5, the columns may be pre-added together to reduce the number of columns. Again, reduction is possible because of the inherent symmetry of the matrix. Thus, column 11 may be added to column 1, column 10 may be added to column 2, and

so on. As a result of the row rise and column rise folding, a 6x6 matrix may be stored in the hardware 16 shown in Figure 5.

Referring to Figure 6, the filter values for a plurality of different filter kernel or matrix sizes are then calculated. In one embodiment, the filters may be calculated in order from the smaller to the larger kernel size. Thus, in the illustrated embodiment, the 6x6 pre-added data storage 26, shown in Figure 6, calculates the 3x3 matrix using row 5, columns 5 and 6 and row 6, columns 5 and 6 and then progressing as illustrated in Figure 6 through the 5x5, 7x7, 9x9, and 11x11 matrixes.

To calculate a given filter, some values in a given diagonal may be added together and accumulated in the adder or accumulator section 28. These results may then be saved for use in filter calculations for the larger kernels.

The calculation of the 3x3 filter utilizes the data in the box labeled 3x3 in storage 26. The pre-added data value sitting in row 6, column 6, is directed by the state machine 11 to the register 36k in section 28.

The pre-added data in diagonal 10 may use one of the AAC section 28 adders 34 and the result is stored in register 36j. In particular, the state machine 11 causes a specified adder, such as the adder 34e, to add the values in diagonal 10 (row 6, column 5 and row 5, column 6) and to place the result in the register 36j, for example. The data value in row 5, column 6, that belongs to diagonal 9 is moved directly to the accumulator storage area 36i, again, because no adding is necessary.

Three multipliers, such as the multipliers 38c, 38d, and 38e, multiply the three data values in the registers 36i-36k with the respective convolution coefficients from the coefficient bank 48. The multiplication results are then added by the adder 42b and that result is added in the adder 36 to the result of the multiplier 38c.

The 5x5 filter, shown in the lower left-hand corner of Figure 6, may be calculated starting at the lower right-hand corner of the data storage 26 and moving up. The data value on the diagonals 11 and 10 are already sitting in their respective accumulators ready for multiplication due to the prior calculation of the 3x3 filter. This is an example of how two filters can be calculated simultaneously. It is evident that the processes used to calculate the 3x3 filter also help to calculate the 5x5 filter.

The values for diagonal 9 need to be added together. One of the data points, row 5, column 5, is sitting in the accumulator section 28. The remaining two rows, in row 4, column 6, and row 6, column 4, need to be accumulated. One of the accumulator section adders may be utilized for this function and the result may be stored in the register 36i.

There are two values for diagonal 8 (row 4, column 5, and row 5, column 4) that are added together and stored in the register 36h. One of the adder accumulator section adders is used to accomplish this task with the result saved in register 36h.

The value for diagonal 7 (row 4, column 4) may be moved into register 36g. The multiplied accumulate section 30 performs the final multiplication step. The values in the registers 36g-36k are multiplied by the corresponding coefficients.

The remaining filters may be calculated the same way. The multiply accumulate section 30 may be used to calculate portions of the filter and subsequently to calculate the remaining data points.

Because many of the steps are done in parallel, the overall number of clock cycles are reduced. The calculations may be pipelined in some embodiments and many steps may be accomplished in one clock cycle.

It is respectfully submitted that the techniques for simultaneously calculating the filter coefficients of different sizes is set forth in detail in the figures and drawings.

Therefore, reconsideration of the rejection is respectfully requested.

Claim 1 was rejected under Section 102 as being anticipated by Park. Hopefully, based on the above discussion, it can be appreciated that all Park does is suggest the possibility of selecting one of a plurality of different kernels. The kernels are simply the arrangement of pixels that might be selected as the data for calculating a filter. But simply suggesting the selection of an appropriate kernel of a plurality of filters in no way suggests simultaneously determining at least two filters of different sizes from that data.

Therefore, reconsideration of the rejection of claim 1, based on the reference to Park, is respectfully requested. On the same analysis, claims 11 and 21 should patentably distinguish over the Park reference.

In view of these remarks, the application is now in condition for allowance and the Examiner's prompt action in accordance therewith is respectfully requested.

Respectfully submitted,

Date: March 12, 2004

Timothy N. Trop, Reg. No. 28,994
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Ste. 100
Houston, TX 77024
713/468-8880 [Phone]
713/468-8883 [Fax]